IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants: Thomas A. Dye; Manuel J. Alvarez II; Peter D. Geiger

Assignee: Quickshift, Inc. (f/k/a Interactive Silicon, Inc.)

Title: Selective Lossless, Lossy, Or No Compression Of Data Base On Address Range, Data Type, And/Or Requesting Agent (As Revised)

Serial No.:	09/239,659	Filing Date:	1/29/99
Examiner:	Hong Chong Kim	Group Art Unit:	2187
Docket No.:	40532-P001M4P1 (f/k/a 5143-01700)	Conf. No.:	6412

Dallas, Texas
May 10, 2004

Mail Stop Appeal Brief - Patents
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

RECEIVED

MAY 1 7 2004

Technology Center 2100

APPEAL BRIEF UNDER 37 C.F.R. § 1.191

Dear Sir:

Appellants submit this Appeal Brief pursuant to the Notice of Appeal filed in this case on December 5, 2003 and received in the Patent Office on or about December 8, 2003. Thus, this Appeal Brief is due February 8, 2004, but extended to May 10, 2004 (since May 8, 2004 was a Saturday) by the petition and fee of $475.00 filed concurrently herewith. Additionally, enclosed is an amount of $165.00, being the amount specified in 37 C.F.R. 1.17(c) for this Appeal Brief. The Commissioner is also authorized to deduct any other amounts required for this Appeal Brief and/or to credit any amounts overpaid to Deposit Account No. 23-2426 (40532-P001M4P1). **This Brief is submitted in triplicate.**

05/14/2004 HWONDAF1 00000027 09239659
01 FC:2402   165.00 OP

I.   REAL PARTY IN INTEREST

The real party in interest is the assignee, Quickshift, Inc., as named in the caption above. Appellants have also enclosed a copy of documents filed separately but concurrently for recording a patent assignment and name change, which reflects that Quickshift, Inc. is the current assignee.

II.   RELATED APPEALS AND INTERFERENCES

Based on information and belief, there are no appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals in the pending appeal. However, Quickshift has appeals pending in the following applications:

> Serial Numbers: 09/915,751
>
> 10/044,786

III.   STATUS OF CLAIMS

Claims 1-3, 5-38, 40-46, 58-70 and 95-122 are pending in the application.

Claims 1-3, 5-25, 37, 38, 40-46, 107 and 108 are allowed.

Claims 26-36, 58-60, 67-70, 95-106 and 109-122 are rejected.

Claims 61-66 are objected to.

IV.   STATUS OF AMENDMENTS

The Appellants' response and claim amendments to the Office Action dated August 1, 2002, having a mailing date of August 5, 2002, have been considered, but the Examiner indicated that they did not place the application in condition for allowance because the Appellants' arguments were deemed unpersuasive.

V.   SUMMARY OF THE INVENTION

The present invention comprises a memory controller which provides improved data efficiency and bandwidth. The memory controller includes a compression/decompression engine, preferably parallel data compression and decompression slices, that are embedded into the memory control logic of the memory controller.

The present invention can selectively use different compression modes, such as lossless, lossy, or no compression. Thus, in addition to the lossless compression/decompression, the memory controller also can include one or more specific lossy compression and decompression modes for particular data formats such as image data, texture maps, digital video and digital audio. The present invention may selectively apply different compression/decompression algorithms depending on one or more of: the type of the data, the requesting agent, or a memory address range.

The present invention also accounts for overflow conditions during compression. Overflow occurs when the data being compressed actually compresses to a larger size than the original data size, or when the data compresses to a smaller size than the original data, but to a larger size than the allocated block size. The present invention handles the overflow case by first determining whether a block will overflow, and second storing an overflow indicator and overflow information with the data. The memory controller preferably generates a header stored with the data that indicates the overflow indicator and overflow information. Thus, the directory information is stored with the data, rather than in separate tables. Compression mode information may also be stored in the header with the data. The present invention thus operates to imbed directory structures directly within the compressed data stream.

The integrated data compression/decompression capabilities of the present invention removes system bottlenecks and increases performance. This allows lower cost systems due to smaller data

storage requirements and reduced bandwidth requirements. This also increases system bandwidth and hence increases system performance. Thus, the memory controller of the present invention is a significant advance over the operation of prior art memory controllers.

VI. ISSUES

A. Is claim 26 properly rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,553,160 to Dawson (hereinafter "Dawson")?

B. Are claims 31 and 35 properly rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,724,582 to Pelanek, et al. (hereinafter "Pelanek")?

C. Are claims 27-30, 95-106 and 109-122 properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Dawson in view of Pelanek?

D. Are claims 32-34 and 36 properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Pelanek in view of Dawson?

E. Are claims 58-60 and 67-70 properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Dawson in view of U.S. Patent No. 5,847,762 to Canfield, et al. (hereinafter "Canfield")?

VII. GROUPING OF THE CLAIMS

Claim 26 forms a first group.

Claims 31 and 35 form a second group.

Claims 27-30, 95-106, and 109-122 form a third group.

Claims 32-34, and 36 form a fourth group.

Claims 58-60, and 67-70 form a fifth group.

Claims 61- 66 form a sixth group.

The reasons for these groupings are set forth in Appellants' arguments in Section VIII.

VIII. ARGUMENT

A. First Group

Claim 26 was rejected under 35 U.S.C. § 102(b) in view of the Dawson patent. Claim 26 recites a system memory which stores data used by a CPU for executing one or more applications. Claim 26 also recites a memory controller coupled to the system memory which performs memory control functions for the system memory, wherein the memory controller includes a compression/decompression engine. Appellants respectfully submit that the cited Dawson patent does not teach or suggest these limitations. Rather, Dawson is directed towards compression for image data only. Dawson's teachings are not directed to compression of other data formats, such as texture maps, digital video, and digital audio, which are within the scope of Appellants' claimed invention. Furthermore, Dawson does not teach or suggest a memory controller which includes a compression/decompression engine, and which performs the operations set forth in claim 26.

The Examiner argues that Fig. 1B and Col. 8, lines 20-30 of Dawson disclose a memory controller including a compression/decompression engine. Appellants respectfully disagree with the Examiner. Fig. 1B specifically shows a compression manager 114, which includes memory controller 150, and separate compression engines 152 and 153. Thus, Fig. 1B specifically illustrates that the compression engines are not part of the memory controller, as is required by Claim 26.

B. Second Group

Claims 31 and 35 were rejected under 35 U.S.C. § 102(b) as being anticipated by Pelanek. Pelanek relates to medical image data which is archived in a recordable optical compact disk.

Serial No. 09/239,659

Pelanek teaches that the medical image data may be stored on the compact disk in two forms: losslessly compressed image data and corresponding lossy compressed image data. *See* Summary of the Invention. According to the teachings of Pelanek: "if the number of losslessly compressed digital medical images constituting a set or case study, exceeds the capacity of a single CD, an entire set of lossy compressed digital medical images will be recorded on each CD of the plurality of CDs which record the case study. Thus, the lossy data is identical on each CD, while the lossless data is appropriately split between multiple CDs." *See* Col. 4, lines 30-37.

Appellants submit that claims 31 and 35 should be allowed over the Pelanek reference because Pelanek does not teach "determining a compression mode for the data based on the one or more destination addresses" (as required by claim 31), or "determining a compression mode for the data based on the data type of the data" (as required by claim 35). Pelanek pertains only to the storage of one type of data: medical image data. Furthermore, Pelanek does not teach determining compression modes based on data type or address ranges, but instead Pelanek teaches storing lossy compressed digital medical images on each disk of a plurality of disks where the medical image data exceeds the capacity of a single disk.

C. Third Group

Claims 27-30, 95-106, and 109-122 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Dawson in view of Pelanek.

Claims 27-30 should be allowed for at least the same reasons discussed above.

With regard to claim 95, this claim recites a method for compressing data which includes "allocating a memory block, wherein the memory block is allocated for uncompressed data" and further includes a step of "storing compressed first data in the allocated memory block." In other words, claim 95 recites a method of storing compressed data in a memory block that was allocated

for uncompressed data. The Examiner has not cited to any teachings in either Dawson or Pelanek which disclose this feature.

Appellants also respectfully submit that the Examiner has not met the Examiner's burden of factually supporting the alleged motivation to combine Dawson and Pelanek. It is the Examiner's burden to factually support any prima facie conclusion of obviousness. The Examiner's duty may not be satisfied by engaging in impermissible hindsight; any conclusion of obviousness must be reached on the basis of facts gleaned from the prior art. The preferred evidence to be offered by the Examiner is an express teaching to modify/combine which is set forth within objectively verifiable sources of prior art. *See* MPEP §§ 2141-2144. In this case, the Examiner has not cited to any express teachings within the Dawson and Pelanek patents which support a motivation to combine these patents to achieve Appellants' claimed invention.

Claims 96-106 which are dependent on claim 95, should be allowed for at least the same reasons.

Claims 109-122 should be allowed for at least the same reasons that claims 26 and 95 should be allowed.

D. Fourth Group

Claims 32-34, and 36 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Pelanek in view of Dawson.

These claims should be allowed for at least the same reasons that claim 35 and 35 should be allowed. Furthermore, again the Examiner has not met the Examiner's burden of factually supporting the alleged motivation to combine the Dawson and Pelanek patents.

E.   Fifth Group

Claims 58-60 and 67-70 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Dawson in view of Canfield.

Claims 58-60 and 67-69 require the step of "determining if the first size of the compressed first data is greater than an allocated memory block size of a first allocated memory block." Appellants submit that Dawson does not teach such a step.   The Examiner alleges that Fig. 4 of Dawson contains such teachings.   However, Fig. 4 and the corresponding text in the specification make it clear that Dawson merely teaches determining whether the size of an image is less than a predetermined value (*e.g.*, 4k bytes).   If the image size is less than 4k bytes, then the image is not compressed. Dawson does not teach, on the other hand, determining if the size of *compressed data is greater than an allocated memory block size.*

Appellants further submit that the Examiner again has not satisfied the Examiner's burden of factually supporting the alleged motivation to combine the Dawson and Canfield patents. Furthermore, the Examiner has not cited to any teachings in Canfield "wherein the header includes an overflow indicator indicating whether the first size of the compressed first data is greater than the allocated memory block size" (as set forth in claim 58).   Instead, Canfield teaches a header that contains signaling information indicating the *type* of compression that was performed on the block. *See* Col. 5, lines 29-35.

For at least these reasons, claim 58-60 and 67-69 should be allowed over the Dawson and Canfield patents.

Claim 70 should be allowed for at least the same reasons that claim 26 should be allowed. Furthermore, again the Examiner has not met the Examiner's burden of factually supporting the alleged motivation to combine the Dawson and Canfield patents. In addition, the Examiner has not

cited to any teachings in Canfield "wherein the header includes compression mode information indicating the compression mode of the first data, *wherein the compression mode information indicates a decompression procedure for decompression of the compressed first data.*" (Emphasis added.) Again, Canfield discloses inserting information indicating the type of compression into a header. However, the Examiner has not cited to any teachings in Canfield wherein the header contains information indicating a decompression procedure for decompression of the compressed first data.

F.   Information Disclosure Statement filed June 6, 2003

Finally, Appellants respectfully request that the Board order the Examiner to consider Appellants' Information Disclosure Statement that they filed on June 6, 2003. In a May 4, 2004 telephone conference with the Examiner, Appellants brought to the Examiner's attention that this Information Disclosure Statement has not yet been considered. The Examiner indicated that he would review the file and report back to Appellants' attorney. Appellants' attorney called again on May 10, 2004 and left the Examiner a voicemail message reminding him of this. However, as of the date of this Appeal Brief, Appellants' attorney has not heard back from the Examiner.

IX.   CONCLUSION

For the above reasons, Appellant respectfully submits that rejection of pending Claims 26-36, 58-60, 61-66, 67-70, 95-106, and 109-122 is unfounded. Accordingly, Appellant requests that the rejection of Claims 26-36, 58-60, 61-66, 67-70, 95-106, and 109-122 be reversed.

This Brief is submitted in triplicate.

Respectfully submitted,

Michael P. Adams
Attorney for Appellant(s)
Reg. No. 34,763

CERTIFICATION UNDER 37 C.F.R. § 1.8

I hereby certify that this correspondence (along with any item referred to as being enclosed herewith) is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on May 10, 2004.

_____
Signature

APPENDIX

26.   A computer system utilizing compressed storage of data, the computer system comprising:

a CPU;

system memory which stores data used by said CPU for executing one or more applications, wherein the system memory also stores an operating system;

a memory controller coupled to said system memory and said CPU, wherein said memory controller performs memory control functions for said system memory, wherein said memory controller includes a compression/decompression engine comprised in said memory controller for compressing and decompressing data transferred to or from said system memory;

wherein the memory controller is operable to:

receive uncompressed data;

determine a compression mode for the data, wherein the compression mode comprises one of lossless compression, lossy compression, or no compression;

selectively compress the uncompressed data, wherein said compressing is selectively performed in response to the compression mode for the data; and

store the data in the memory.

27.   The computer system of claim 26,

wherein the compression mode is determined in response to one or more of: a requesting agent which provides the data; an address range where the data is stored; and/or a data type of the data.

28.   The computer system of claim 26,

wherein the memory controller is operable to receive one or more destination addresses indicating a storage destination for the data in the memory;

wherein the memory controller is operable to analyze the one or more destination addresses to determine the compression mode.

29. The computer system of claim 26,

wherein the uncompressed data is received from a requesting agent; and

    wherein the memory controller determines the compression mode based on the

        requesting agent.

30. The computer system of claim 26, wherein the data has a data type;

    wherein the memory controller determines the compression mode for the data based

        on the data type of the data.

31. A method for compressing data and storing the compressed data in a memory in a computer system, the method comprising:

    receiving uncompressed data;

    receiving one or more destination addresses indicating a storage destination for the data in the

        memory;

    determining a compression mode for the data based on the one or more destination addresses;

    selectively compressing the uncompressed data, wherein said compressing is selectively

        performed in response to the compression mode for the data; and

        storing the data in the memory at the one or more destination addresses.

32. The method of claim 31, wherein the compression mode comprises one of lossless compression, lossy compression, or no compression;

    wherein, in said selectively compressing:

        the data is compressed using a lossless compression format if said compression mode

            indicates lossless compression for the data;

        the data is compressed using a lossy compression format if said compression mode

            indicates lossy compression for the data; and

        the data is not compressed if said compression mode indicates no compression for the

            data.

33. A method for compressing data and storing the compressed data in a memory in a computer system, the method comprising:

    receiving uncompressed data from a requesting agent;

        Serial No. 09/239,659

determining a compression mode for the data based on the requesting agent;

selectively compressing the uncompressed data, wherein said compressing is selectively

performed in response to the compression mode for the data; and

storing the data in the memory at the one or more destination addresses.

34.   The method of claim 33, wherein the compression mode comprises one of lossless compression, lossy compression, or no compression;

wherein, in said selectively compressing:

the data is compressed using a lossless compression format if said compression mode

indicates lossless compression for the data;

the data is compressed using a lossy compression format if said compression mode

indicates lossy compression for the data; and

the data is not compressed if said compression mode indicates no compression for the

data.

35.   A method for compressing data and storing the compressed data in a memory in a computer system, the method comprising:

receiving uncompressed data, wherein the data has a data type;

determining a compression mode for the data based on the data type of the data;

selectively compressing the uncompressed data, wherein said compressing is selectively

performed in response to the compression mode for the data; and

storing the data in the memory at the one or more destination addresses.

36.   The method of claim 35, wherein the compression mode comprises one of lossless compression, lossy compression, or no compression;

wherein, in said selectively compressing:

the data is compressed using a lossless compression format if said compression mode

indicates lossless compression for the data;

the data is compressed using a lossy compression format if said compression mode

indicates lossy compression for the data; and

the data is not compressed if said compression mode indicates no compression for the

data.

57.   The method of claim 54, further comprising:

decompressing the compressed first data to produce uncompressed first data, wherein said

decompressing comprises:

analyzing the compression mode information stored with the data;

determining a decompression procedure based on the compression mode information

stored with the data; and

decompressing the compressed first data using the determined decompression

procedure.

58.   A method for compressing data and storing the compressed data in a memory in a

computer system, the method comprising:

receiving uncompressed first data;

compressing the uncompressed first data to produce compressed first data, wherein said

compressed first data has a first size;

determining if the first size of the compressed first data is greater than an allocated memory

block size of a first allocated memory block;

creating a header, wherein the header includes an overflow indicator indicating whether the

first size of the compressed first data is greater than the allocated memory block size;

and

storing the compressed first data and the header in the memory.

59.   The method of claim 58, wherein said determining determines that the first size of the

compressed first data is less than or equal to the allocated memory block size;

wherein the overflow indicator indicates that the first allocated memory block stores

all of the compressed first data.

60.   The method of claim 59, wherein said overflow indicator indicates that the last symbol

of the compressed first data is stored in the first allocated memory block.

61. The method of claim 58, wherein said determining determines that the first size of the compressed first data is greater than the allocated memory block size;

    wherein the overflow indicator indicates that the first allocated memory block does not store all of the compressed first data;

    the method further comprising:

        allocating a first overflow memory block;

        storing overflow information in the header, wherein the overflow information includes an overflow address pointer which points to the first overflow memory block;

        wherein said storing comprises:

            storing a first portion of the compressed first data and the header in the first allocated memory block; and

        storing an overflow portion of the compressed first data in the first overflow memory block.

62. The method of claim 61, wherein the first overflow memory block has a fixed size.

63. The method of claim 61, further comprising:

determining whether the overflow portion has a size greater than the first overflow memory block;

creating an overflow header, wherein the overflow header includes an overflow indicator indicating whether the overflow portion has a size greater than the first overflow memory block;

    wherein said storing the overflow portion includes storing the overflow portion and the overflow header in the first overflow memory block.

64. The method of claim 63, further comprising:

wherein said determining determines that the overflow portion of the compressed first data has a size greater than the first overflow memory block;

wherein the overflow indicator in the overflow header indicates that the first overflow memory block does not store all of the overflow portion;

the method further comprising:

allocating a second overflow memory block in response to determining that the overflow portion of the compressed first data is greater than the first overflow memory block;

storing overflow information in the first overflow header, wherein the overflow information includes an overflow address pointer which points to the second overflow memory block;

wherein said storing comprises:

storing a first portion of the compressed first data and the header in the first allocated memory block;

storing a first overflow portion of the compressed first data in the first overflow memory block; and

storing a second overflow portion of the compressed first data in the second overflow memory block.

65.   The method of claim 58, wherein said determining determines that the first size of the compressed first data is greater than the allocated memory block size;

wherein the overflow indicator indicates that the first allocated memory block does not store all of the compressed first data;

the method further comprising:

allocating a plurality of overflow memory blocks, including a first overflow memory block and a last overflow memory block;

storing overflow information in the header, wherein the overflow information includes an overflow address pointer which points to a first overflow memory block;

wherein said storing comprises:

storing a first portion of the compressed first data and the header in the first allocated memory block; and

for each of the overflow memory blocks except the last overflow memory block, storing, in the respective overflow memory block, an overflow portion of the compressed first data and a header pointing to a subsequent overflow memory block.

66.   The method of claim 58, wherein said determining determines that the first size of the compressed first data is greater than the allocated memory block size;

wherein the overflow indicator indicates that the first allocated memory block does not store

all of the compressed first data;

the method further comprising:

allocating one or more overflow memory blocks, wherein the first allocated memory

block and the one or more overflow memory blocks are insufficient to store the

compressed first data;

generating an interrupt to a driver in response to the first allocated memory block and

the one or more overflow memory blocks being insufficient to store the

compressed first data;

the driver allocating additional overflow memory blocks in response to the interrupt.

67.   The method of claim 58, wherein said determining determines if the first size of the compressed first data and a maximum header size are greater than the allocated memory block size.

68.   The method of claim 58, further comprising:

allocating the first allocated memory block in response to receiving the uncompressed

first data, wherein the first allocated memory block is allocated according to a

pre-determined compression ratio.

69.   The method of claim 58, wherein the computer system includes an operating system, the method further comprising:

the operating system allocating the first allocated memory block in response to

receiving the uncompressed first data.

70.   A computer system including a memory controller having an embedded compression/decompression engine, the computer system comprising:

a CPU;

system memory which stores data used by said CPU for executing one or more applications;

a memory controller coupled to said system memory and said CPU, wherein said memory

controller performs memory control functions for said system memory, wherein said

memory controller includes said compression/decompression engine comprised in said memory controller for compressing and decompressing data transferred to or from said system memory;

wherein said memory controller is operable to:

receive uncompressed first data;

selectively compress the uncompressed first data to produce compressed first data according to a compression mode;

create a header, wherein the header includes compression mode information indicating the compression mode of the first data, wherein the compression mode information indicates a decompression procedure for decompression of the compressed first data; and

store the compressed first data and the header in the memory.

95.   A method for compressing data and storing the compressed data in a memory in a computer system, the method comprising:

allocating a memory block, wherein the memory block is allocated for uncompressed data;

receiving uncompressed first data;

receiving one or more destination addresses indicating a storage destination of the first data in the allocated memory block;

compressing the uncompressed first data to produce compressed first data;

storing the compressed first data in the allocated memory block at the one or more destination addresses.

96.   The method of claim 95, wherein said storing does not perform address translation of the one or more destination addresses, wherein said storing provides reduced latency.

97.   The method of claim 95, wherein the uncompressed first data has a first size, wherein the compressed first data has a second smaller size;

wherein said storing does not perform address translation of the one or more destination addresses, wherein said storing does not perform memory minimization.

98. The method of claim 95, wherein the computer system includes an operating system, wherein the operating system allocates the memory block for uncompressed data;

wherein the operating system does not account for the compression operation.

99. The method of claim 95, wherein the computer system includes an operating system, wherein the operating system allocates the memory block for uncompressed data;

wherein the operating system is unaware of the compression operation.

100. The method of claim 95, wherein the compressed first data occupies a first portion of the allocated memory block, the method further comprising:

allocating a portion of the allocated memory block as overflow storage.

101. The method of claim 100,

wherein the uncompressed first data comprises a plurality of blocks each having an original size, wherein one or more of the blocks compress to a larger size than the original size;

wherein said storing the compressed first data includes storing overflow data in the portion of the allocated memory block allocated as overflow storage.

102. The method of claim 95, wherein the uncompressed first data comprises application data generated by a CPU in the computer system.

103. The method of claim 95, wherein the memory comprises a system memory which stores application data generated by a CPU in the computer system.

104. The method of claim 95, further comprising

receiving a request for the first data;

decompressing the compressed first data to produce uncompressed first data;

providing the uncompressed first data in response to the request.

105. The method of claim 95, further comprising

receiving a request for the first data, wherein the request includes the one or more destination addresses in the allocated memory block where the compressed first data is stored;

accessing the compressed first data from the memory using the one or more destination addresses;

decompressing the compressed first data to produce uncompressed first data; and

providing the uncompressed first data in response to the request.

106.   The method of claim 95, wherein the computer system includes a memory controller, wherein the memory controller performs said receiving uncompressed first data, said receiving one or more destination addresses, said compressing the uncompressed first data to produce compressed first data, and said storing the compressed first data.

109.   A computer system utilizing compressed storage of data, the computer system comprising:

a CPU;

system memory which stores data used by said CPU for executing one or more applications, wherein the system memory also stores an operating system;

a memory controller coupled to said system memory and said CPU, wherein said memory controller performs memory control functions for said system memory, wherein said memory controller includes a compression/decompression engine comprised in said memory controller for compressing and decompressing data transferred to or from said system memory;

wherein memory blocks are allocated in the system memory for uncompressed data;

wherein the memory controller is operable to:

receive uncompressed first data;

receive one or more destination addresses indicating a storage destination of the first data in an allocated memory block;

compress the uncompressed first data to produce compressed first data; and

store the compressed first data in the allocated memory block at the one or more destination addresses.

110.   The computer system of claim 109, wherein, in storing the compressed first data, the memory controller does not perform address translation of the one or more destination addresses, wherein the memory controller provides reduced latency.

111. The computer system of claim 109, wherein the uncompressed first data has a first size, wherein the compressed first data has a second smaller size;

    wherein the memory controller does not perform address translation of the one or more destination addresses, wherein the memory controller does not perform memory minimization.

112. The computer system of claim 109, wherein the computer system includes an operating system, wherein the operating system allocates the memory block for uncompressed data; wherein the operating system does not account for the compression operation.

113. The computer system of claim 109, wherein the computer system includes an operating system, wherein the operating system allocates the memory block for uncompressed data; wherein the operating system is unaware of the compression operation.

114. The computer system of claim 109, wherein the compressed first data occupies a first portion of the allocated memory block;

    wherein the memory controller is operable to allocate a portion of the allocated memory block as overflow storage.

115. The computer system of claim 114,

wherein the uncompressed first data comprises a plurality of blocks each having an original size, wherein one or more of the blocks compress to a larger size than the original size;

    wherein the memory controller is operable to store overflow data in the portion of the allocated memory block allocated as overflow storage.

116. The computer system of claim 109, wherein the uncompressed first data comprises application data generated by the CPU.

117. The computer system of claim 109, wherein the memory controller is further operable to:

receive a request for the first data;

decompress the compressed first data to produce uncompressed first data; and

    provide the uncompressed first data in response to the request.

    Serial No. 09/239,659

118. The computer system of claim 109, wherein the memory controller is further operable to:

receive a request for the first data, wherein the request includes the one or more destination addresses in the allocated memory block where the compressed first data is stored;

access the compressed first data from the system memory using the one or more destination addresses;

decompress the compressed first data to produce uncompressed first data; and

provide the uncompressed first data in response to the request.

119. A method for compressing data and storing the compressed data in a memory in a computer system, the method comprising:

allocating a memory block, wherein the memory block is allocated according to a pre-determined compression ratio;

receiving uncompressed first data;

receiving one or more destination addresses indicating a storage destination of the first data in the allocated memory block;

compressing the uncompressed first data to produce compressed first data; and

storing the compressed first data in the allocated memory block at the one or more destination addresses.

120. The method of claim 119, wherein said storing includes performing address translation of the one or more destination addresses, wherein said address translation minimizes memory usage.

121. The method of claim 119, wherein the computer system includes an operating system, wherein the operating system allocates the memory block for uncompressed data according to the pre-determined compression ratio.

122. The method of claim 119, wherein the uncompressed first data has a first size, wherein the compressed first data has a second smaller size;

the method further comprising:

determining if the compressed first data fits within the allocated memory block; and

allocating an overflow memory block if the compressed first data does not fit within the allocated memory block.

Serial No. 09/239,659